Fast Marginalized Block SBL Algorithm

Benyuan Liu[1,*], Zhilin Zhang[2], Hongqi Fan[1], Zaiqi Lu[1], Qiang Fu[1]

arXiv:1211.4909v4 [cs.IT] 22 Jan 2013

Abstract

The performance of sparse signal recovery can be improved if both sparsity and correlation structure of signals can be exploited. One typical correlation structure is intra-block correlation in block sparse signals. To exploit this structure, a framework, called block sparse Bayesian learning (BSBL) framework, has been proposed recently. Algorithms derived from this framework showed superior performance but they are not very fast, which limits their applications. This work derives an efficient algorithm from this framework, using a marginalized likelihood maximization method. Compared to existing BSBL algorithms, it has close recovery performance but is much faster. Therefore, it is more suitable for large scale datasets and applications requiring real-time implementation.

Keywords:

Sparse Signal Recovery; Compressed Sensing; Block Sparse Bayesian Learning (BSBL); Intra-Block Correlation; Fast Marginal Likelihood Maximization.

1. Introduction

Sparse signal recovery and the associated compressed sensing [1] can recover a signal with small number of measurements with high probability of successes (or sufficient small errors), given that the signal is sparse or can be sparsely represented in some domain. It has been found that exploiting structure information of a signal can further improve the recovery performance. In practice, a signal generally has rich structures. One structure is

*liubenyuan@gmail.com

[1]Department of Electrical and Communication Engineering, University of Defense Technology, Changsha, Hunan, 410074, China.

[2]The Department of Electrical and Computer Engineering, University of California, San Diego, La Jolla, CA 92093-0407, USA.

the block/group sparse structure[2, 3, 4, 5], which refers to the case when nonzero entries of a signal all cluster around some locations. Existing algorithms exploit such information showed improved recovery performance.

Recently, noticing intra-block correlation widely exists in real-world signals, Zhang and Rao [2, 6] proposed the block sparse Bayesian learning (BSBL) framework. A number of algorithms have been derived from this framework, and showed superior ability to recover block sparse signals or even non-sparse signals [7]. But these BSBL algorithms are not fast, and thus cannot be applied to large-scale datasets.

In this work, we propose a BSBL algorithm using a fast marginalized likelihood maximization (FMLM) method [8]. Thanks to the BSBL framework, it can exploit both block structure and intra-block correlation. Experiments conducted on both synthetic data and real ECG data showed that the proposed algorithm significantly outperforms traditional algorithms which only exploit block structure such as Model-CoSaMP[4] and Block-OMP[9], and has similar recovery accuracy as BSBL algorithms [2]. However, it is much faster than the BSBL algorithms[2] and thus is more suitable for large scale problems.

Throughout the paper, **Bold** symbols are reserved for vectors and matrices.

2. The Block Sparse Bayesian Learning Framework

A block sparse signal $\mathbf{x}$ has the following structure:

$$\mathbf{x} = [\underbrace{x_1, \cdots, x_{d_1}}_{\mathbf{x}_1^T}, \cdots, \underbrace{x_1, \cdots, x_{d_g}}_{\mathbf{x}_g^T}]^T, \tag{1}$$

which means $\mathbf{x}$ has g blocks, and only a few blocks are nonzero. Here d_i is the block size for the ith block. To model the intra-block correlation in the i-th block, the BSBL framework suggests to use the parameterized Gaussian distribution:

$$p(\mathbf{x}_i; \gamma_i, \mathbf{B}_i) = \mathcal{N}(\mathbf{x}_i; \mathbf{0}, \gamma_i \mathbf{B}_i). \tag{2}$$

with unknown deterministic parameters γ_i and $\mathbf{B}_i$. Although blocks have intra-block correlation, the framework assumes that blocks are mutually independent. The observation $\mathbf{y}$ is

obtained by

$$\mathbf{y} = \mathbf{\Phi}\mathbf{x} + \mathbf{n}, \tag{3}$$

where $\mathbf{\Phi}$ is a $M \times N$ sensing matrix and $\mathbf{n}$ is the observation noise. The observation noise is assumed to be independent and Gaussian with zero mean and variance equal to β^{-1}. β is also unknown. Thus the likelihood is given by

$$p(\mathbf{y}|\mathbf{x};\beta) = \mathcal{N}(\mathbf{\Phi}\mathbf{x}, \beta^{-1}\mathbf{I}). \tag{4}$$

The posterior can be calculated as $p(\mathbf{x}|\mathbf{y};\{\gamma_i,\mathbf{B}_i\}_i,\beta) = \mathcal{N}(\boldsymbol{\mu},\mathbf{\Sigma})$ with $\mathbf{\Sigma} \triangleq (\mathbf{\Gamma}^{-1}+\mathbf{\Phi}^T\beta\mathbf{\Phi})^{-1}$ and $\boldsymbol{\mu} \triangleq \mathbf{\Sigma}\mathbf{\Phi}^T\beta\mathbf{y}$, where $\mathbf{\Gamma}$ denotes a block diagonal matrix with the ith principal block given by $\gamma_i\mathbf{B}_i$. Once all the parameters, namely $\{\gamma_i,\mathbf{B}_i\}_i,\beta$, are estimated, the MAP estimate of the signal $\mathbf{x}$ can be directly obtained from the mean of the posterior, i.e.,

$$\mathbf{x} = \mathbf{\Sigma}\mathbf{\Phi}^T\beta\mathbf{y}. \tag{5}$$

To estimate the parameters, the following cost function is generally used, which is derived from the Type II maximum likelihood [2]:

$$\mathcal{L}(\{\gamma_i,\mathbf{B}_i\}_i,\beta) = \log|\mathbf{C}| + \mathbf{y}^T\mathbf{C}^{-1}\mathbf{y}, \tag{6}$$

where $\mathbf{C} = \beta^{-1}\mathbf{I} + \mathbf{\Phi}\mathbf{\Gamma}\mathbf{\Phi}^T$. There are several methods to minimize the cost function [2]. In the following we consider to use the marginalized likelihood maximization method, which was used by Tipping et al. [8] for their basic SBL algorithm and later was used by Ji et al. [10] for their Bayesian compressive sensing algorithm.

3. The Proposed Fast Block SBL Algorithm

3.1. The Main-body of the Algorithm

The cost function (6) can be optimized in a block way. We denote by $\mathbf{\Phi}_i$ the ith block in $\mathbf{\Phi}$ with the column indexes corresponding to the ith block of the signal $\mathbf{x}$. Then $\mathbf{C}$ can be rewritten as:

$$\mathbf{C} = \beta^{-1}\mathbf{I} + \sum_{m\neq i}\mathbf{\Phi}_m\gamma_m\mathbf{B}_m\mathbf{\Phi}_m^T + \mathbf{\Phi}_i\gamma_i\mathbf{B}_i\mathbf{\Phi}_i^T, \tag{7}$$

$$= \mathbf{C}_{-i} + \mathbf{\Phi}_i\gamma_i\mathbf{B}_i\mathbf{\Phi}_i^T, \tag{8}$$

where $\mathbf{C}_{-i} \triangleq \beta^{-1}\mathbf{I} + \sum_{m \neq i} \mathbf{\Phi}_m \gamma_m \mathbf{B}_m \mathbf{\Phi}_m^T$. Using the Woodbury Identity,

$$|\mathbf{C}| = |\gamma_i \mathbf{B}_i||\mathbf{C}_{-i}||\mathbf{A}_i^{-1} + \mathbf{s}_i|, \tag{9}$$

$$\mathbf{C}^{-1} = \mathbf{C}_{-i}^{-1} - \mathbf{C}_{-i}^{-1}\mathbf{\Phi}_i(\mathbf{A}_i^{-1} + \mathbf{s}_i)^{-1}\mathbf{\Phi}_i^T\mathbf{C}_{-i}^{-1}, \tag{10}$$

where $\mathbf{A}_i \triangleq \gamma_i \mathbf{B}_i$, $\mathbf{s}_i \triangleq \mathbf{\Phi}_i^T \mathbf{C}_{-i}^{-1} \mathbf{\Phi}_i$, and $\mathbf{q}_i \triangleq \mathbf{\Phi}_i^T \mathbf{C}_{-i}^{-1}\mathbf{y}$, the Equation (6) can be rewritten as:

$$\begin{aligned}\mathcal{L} =& \log|\mathbf{C}_{-i}| + \mathbf{y}^T\mathbf{C}_{-i}^{-1}\mathbf{y} \\ &+ \log|\mathbf{I}_{d_i} + \mathbf{A}_i\mathbf{s}_i| - \mathbf{q}_i^T(\mathbf{A}_i^{-1} + \mathbf{s}_i)^{-1}\mathbf{q}_i, \qquad (11)\\ =& \mathcal{L}(-i) + \mathcal{L}(i), \qquad (12)\end{aligned}$$

where $\mathcal{L}(-i) \triangleq \log|\mathbf{C}_{-i}| + \mathbf{y}^T\mathbf{C}_{-i}^{-1}\mathbf{y}$, and

$$\mathcal{L}(i) = \log|\mathbf{I}_{d_i} + \mathbf{A}_i\mathbf{s}_i| - \mathbf{q}_i^T(\mathbf{A}_i^{-1} + \mathbf{s}_i)^{-1}\mathbf{q}_i, \tag{13}$$

which only depends on γ_i and $\mathbf{B}_i$.

Setting $\frac{\partial \mathcal{L}(i)}{\partial \gamma_i} = 0$, we have the updating rule

$$\gamma_i = \frac{1}{d_i}\mathrm{Tr}\left[\mathbf{B}_i^{-1}\mathbf{s}_i^{-1}(\mathbf{q}_i\mathbf{q}_i^T - \mathbf{s}_i)\mathbf{s}_i^{-1}\right]. \tag{14}$$

Setting $\frac{\partial \mathcal{L}(i)}{\partial \mathbf{B}_i} = \mathbf{0}$, we have the updating rule

$$\mathbf{B}_i = \frac{\mathbf{s}_i^{-1}(\mathbf{q}_i\mathbf{q}_i^T - \mathbf{s}_i)\mathbf{s}_i^{-1}}{\gamma_i}. \tag{15}$$

3.2. Regularization to $\mathbf{B}_i$

As noted in [2], regularization to $\mathbf{B}_i$ is required due to limited data. It has been shown [6] that in noiseless cases the regularization does not affect the global minimum of the cost function (6), i.e., the global minimum still corresponds to the true solution; the regularization only affects the probability of the algorithm to converge to the local minima. A good regularization can largely reduce the probability of local convergence. Although theories on regularization strategies are lacking, some empirical methods [6, 2] were presented.

One is to model the entries in each block as a first-order Auto-Regressive (AR) process with the AR coefficient r_i. As a result, $\mathbf{B}_i$ has the following form

$$\mathbf{B}_i = \mathrm{Toeplitz}([1, r_i, \cdots, r_i^{d_i-1}]). \tag{16}$$

where Toeplitz(·) is a Matlab command expanding a real vector into a symmetric Toeplitz matrix. Thus the correlation level of the intra-block correlation is reflected by the value of r_i. r_i can be estimated from the cost function (6) directly, or can be empirically roughly calculated from the estimated $\mathbf{B}_i$ in (15) as shown in [2]. We used the latter method, since it provides satisfactory results and saves lots of computation. According to [2], r_i is calculated by $r_i \triangleq \frac{m_1^i}{m_0^i}$, where m_0^i (res. m_1^i) is the average of entries along the main diagonal (res. the main sub-diagonal) of the matrix $\mathbf{B}_i$. This calculation cannot ensure r_i has a feasible value, i.e. $|r_i| < 0.99$. Thus in practice, we calculate r_i by

$$r_i = \text{sign}(\frac{m_1^i}{m_0^i}) \min\left\{\left|\frac{m_1^i}{m_0^i}\right|, 0.99\right\}. \tag{17}$$

Our algorithm using this regularization (16)-(17) is denoted by **BSBL-FM(1)**.

In many real-world applications, the intra-block correlation in each block of a signal tends to be positive and high together. Thus, one can further constrain that all the intra-block correlation values of blocks have the same AR coefficient r [2],

$$r = \frac{1}{g}\sum_{i=1}^{g} r_i, \tag{18}$$

where r is the average of all the r_i. Then, $\mathbf{B}_i$ is reconstructed as

$$\mathbf{B}_i = \text{Toeplitz}([1, r, \cdots, r^{d_i-1}]). \tag{19}$$

Our algorithm using this regularization (18)-(19) is denoted by **BSBL-FM(2)**.

3.3. Remarks on β

The parameter β^{-1} is the noise variance in our model. It can be estimated by a number of methods. However, the resulting updating rule is generally not robust and requires some regularization [2, 6]. In practice, people treat it as a regularizer and assign some specific values to it [3]. Similar to [10], we select $\beta = 10^{-6}$ in noiseless simulations, $\beta = 0.1\|y\|_2^2$ in general noisy scenarios (e.g. SNR < 20 dB), and $\beta = 0.01\|y\|_2^2$ in high SNR scenarios (e.g. SNR ≥ 20 dB).

[3] For example, one can see this by examining the published codes of the algorithms in [10, 5].

4. Experiments

In the experiments [4]both our BSBL-FM(1) and BSBL-FM(2) were used. Our algorithm was also performed in the way that it ignored intra-block correlation, i.e., fixing $\mathbf{B}_i = \mathbf{I}(\forall i)$ (denoted by BSBL-FM(0)). For comparison, two BSBL algorithms, i.e., BSBL-BO and BSBL-ℓ_1 [2], were used (BSBL-ℓ_1 used the Group Basis Pursuit [11] in its inner loop). Besides, a variational inference based SBL algorithm which also exploits the intra-block correlation (denoted by VBGS) [5] was selected. It used its default parameters. Model-CoSaMP[4] and Block-OMP[9] (given the true sparsity) were used as the benchmark in noiseless situations, while the Group Basis Pursuit [11] was used as the benchmark in noisy situations.

The performance indexes were the normalized mean square error (NMSE) in noisy situations and the success rate in noiseless situations. The NMSE was defined as $\|\hat{\mathbf{x}} - \mathbf{x}_{gen}\|_2^2/\|\mathbf{x}_{gen}\|_2^2$, where $\hat{\mathbf{x}}$ was the estimate of the true signal $\mathbf{x}_{gen}$. The success rate was defined as the percentage of successful trials in total experiments (A successful trial was defined the one when NMSE$\leq 10^{-5}$).

In all the experiments except for the last one, the sensing matrix was a random Gaussian matrix, and it was generated in each trial of each experiment. The computer used in the experiments had 2.5GHz CPU and 2G RAM.

4.1. Empirical Phase Transition

In the first experiment, we studied the phase transitions of all the algorithms in exact recovery of block sparse signals in noiseless situations. The phase transition curve [12] is to show how the success rate is affected by the sparsity level (defined as $\rho = K/M$, where K is the total number of non-zero elements) and indeterminacy (defined as $\delta = M/N$).

The generated signal consisted of 20 blocks with the identical block size 25. The number of non-zero blocks varied from 1 to 10 while their locations were determined randomly. Each non-zero block was generated by a multivariate Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{\text{gen}})$ with

[4]The source codes are available: `http://nudtpaper.googlecode.com/files/bsbl_fm.zip`

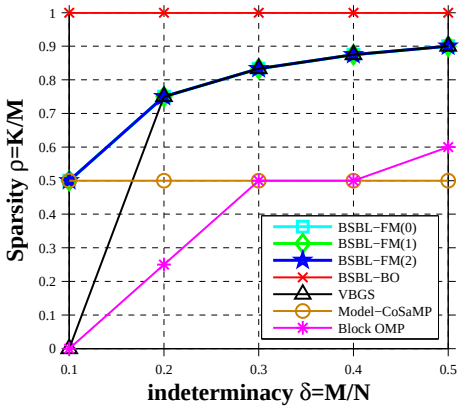

Figure 1: Empirical 96% phase transitions of all algorithms. Each point on the plotted phase transition curve corresponds to the success rate larger than or equal to 0.96. Above the curve the success rate sharply drops.

$\mathbf{\Sigma}_{\text{gen}} \triangleq \text{Toeplitz}([1, r, \cdots, r^{24}])$. The parameter r, which reflected the intra-block correlation level, was set to 0.95. The number of measurements varied from $M = 50$ to $M = 250$.

The results averaged over 100 trials are shown in Fig. 1. Both BSBL-FM and BSBL-BO showed impressive phase transition performance. We see that as a greedy method, BSBL-FM performed better than VBGS, Model-CoSaMP and Block-OMP.

4.2. Performance in Noisy Environments

In each trial we generated a signal of the length $N = 512$. It consisted of 32 blocks with the block size 16. Among them 5 blocks were non-zero. The intra-block correlation level, i.e., r, of each block (generated as in Section 4.1) was uniformly chosen from 0.8 to 0.99. The number of measurement was 128. The SNR, defined as $\text{SNR(dB)} \triangleq 20\log_{10}(\|\mathbf{\Phi}\mathbf{x}_{gen}\|_2/\|\mathbf{n}\|_2)$, varied from 10dB to 25dB. In this experiment we also calculated the oracle result, which was the least square estimate of $\mathbf{x}_{gen}$ given the true support.

From the results (Fig. 2), we see that the proposed algorithm when exploiting intra-block correlation outperformed Group Basis Pursuit and VBGS, and had close recover performance as BSBL-BO, BSBL-ℓ_1. The BSBL-FM(1), BSBL-FM(2) and BSBL-BO even outperformed the oracle estimate, this may be due to that the oracle property utilized only the true support information while ignored the structure in signals (i.e., the intra-block correlation).

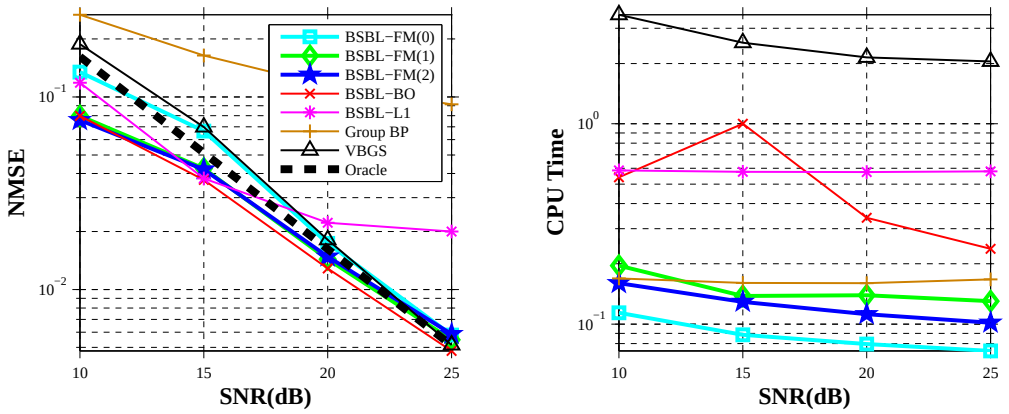

Figure 2: The comparison in NMSE and CPU Time of all the algorithms under different SNR levels.

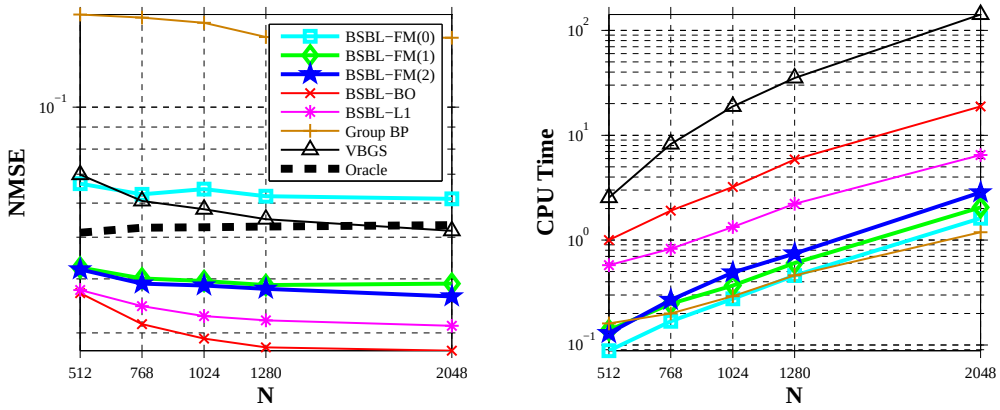

Figure 3: The comparison in NMSE and CPU Time with varying N.

It is worthy noticing that our algorithm had the fastest speed.

4.3. Performance in Larger N

This experiment was designed to show the advantage of our algorithm in speed. The signal consisted of 32 blocks with identical block size, five of which were randomly located non-zero blocks. The length of the signal, N, was varied from 512 to 2048. SNR=15 dB.

The results (Fig. 3) show that the proposed algorithm, although the recovery performance was slightly poorer than BSBL-BO and BSBL-ℓ_1, had the obvious advantage in speed. This implies that the proposed algorithm may be a better choice for large-scale problems. Also, by comparing BSBL-FM(1) and BSBL-FM(2) to BSBL-FM(0), we can see its recovery

performance was improved due to the exploitation of intra-block correlation.

4.4. Application to Telemonitoring of Fetal Electrocardiogram

Fetal electrocardiogram (FECG) telemonitoring via low energy wireless body-area networks [7] is an important approach to monitor fetus health state. BSBL, as an important branch of compressed sensing, has shown great promising in this application[7]. Using BSBL, one can compress raw FECG recordings using a sparse binary matrix, i.e.,

$$\mathbf{y} = \mathbf{\Phi x} \tag{20}$$

where $\mathbf{x}$ is a raw FECG recording, $\mathbf{\Phi}$ is the sparse binary matrix, and $\mathbf{y}$ is the compressed data. It have been showed[13] that using a sparse binary matrix as the sensing matrix can greatly reduce the energy consumption while achieving competitive compression ratio. Then $\mathbf{y}$ is sent to a remote computer. In this computer BSBL algorithms can recover the raw FECG recordings with high accuracy such that independent component analysis (ICA) decomposition [14] on the recovered recordings keeps high fidelity (and a clean FECG is presented after the ICA decomposition).

Here we repeated the experiment in Section III.B in [7] [5]. using the same dataset, the same sensing matrix (a sparse binary matrix with the size 256×512 and each column consisting of 12 entries of 1s with random locations), and the same block partition ($d_i = 24(\forall i)$).

Since it has been shown [7] that non-SBL algorithms failed in this application due to the non-sparsity of raw FECG recordings and the less-sparsity of their representation coefficients in most transform domains, we only compared our algorithm BSBL-FM with VBGS and BSBL-BO. BSBL-BO recovered raw FECG recordings directly as shown in [7]. Differently, BSBL-FM and VBGS first recovered the discrete cosine transform (DCT) coefficients $\boldsymbol{\theta}$ of the recordings according to

$$\mathbf{y} = (\mathbf{\Phi D})\boldsymbol{\theta} \tag{21}$$

[5]The experiment demo is available: `https://sites.google.com/site/researchbyzhang/bsbl`

Table 1: NMSE and the total CPU time in recovery of the whole FECG recordings.

	Average NMSE	Total CPU Time(s)
BSBL-BO	0.0028	712.9
BSBL-FM(2)	0.0035	175.9
BSBL-FM(1)	0.0037	178.7
VBGS	0.1126	9670.8

using $\mathbf{y}$ and $\mathbf{\Phi D}$, where $\mathbf{D}$ was the basis of the DCT transform such that $\mathbf{x} = \mathbf{D}\boldsymbol{\theta}$. Then they reconstructed the original raw FECG recordings according to $\mathbf{x} = \mathbf{D}\boldsymbol{\theta}$ using $\mathbf{D}$ and $\boldsymbol{\theta}$.

The NMSE measured on the recovered FECG recordings is shown in Tab. 1. We can see although BSBL-FM had slightly poorer recovery accuracy than BSBL-BO, it had much faster speed. In fact, the ICA decomposition on the recovered recordings by BSBL-FM also presented a clean FECG (see Fig. 4), and the decomposition was almost the same as the ICA decomposition on the original recordings. In this experiment we noticed that VBGS took long time to recover the FECG recordings, and had the largest NMSE. Besides, the ICA decomposition on its recovered recordings didn't present the clean FECG. This reason may be due to the fact that the DCT coefficients of the raw FECG recordings are not sufficiently sparse, and recovering these less-sparse coefficients is very difficult for non-BSBL algorithms[7].

5. Conclusion

The block sparse Bayesian learning (BSBL) algorithms have superior performance than other state-of-the-art algorithms in recovery of block sparse signals, especially when the signals have intra-block correlation. However, existing BSBL algorithms are not very fast. Thus they may not be suitable for large-scale problems. To fill this gap, this work proposed a fast BSBL algorithm, which also exploits the intra-block correlation. Experiments showed that it significantly outperforms non-BSBL algorithms, and has close recovery performance as existing BSBL algorithms, but is the fastest among these BSBL algorithms.

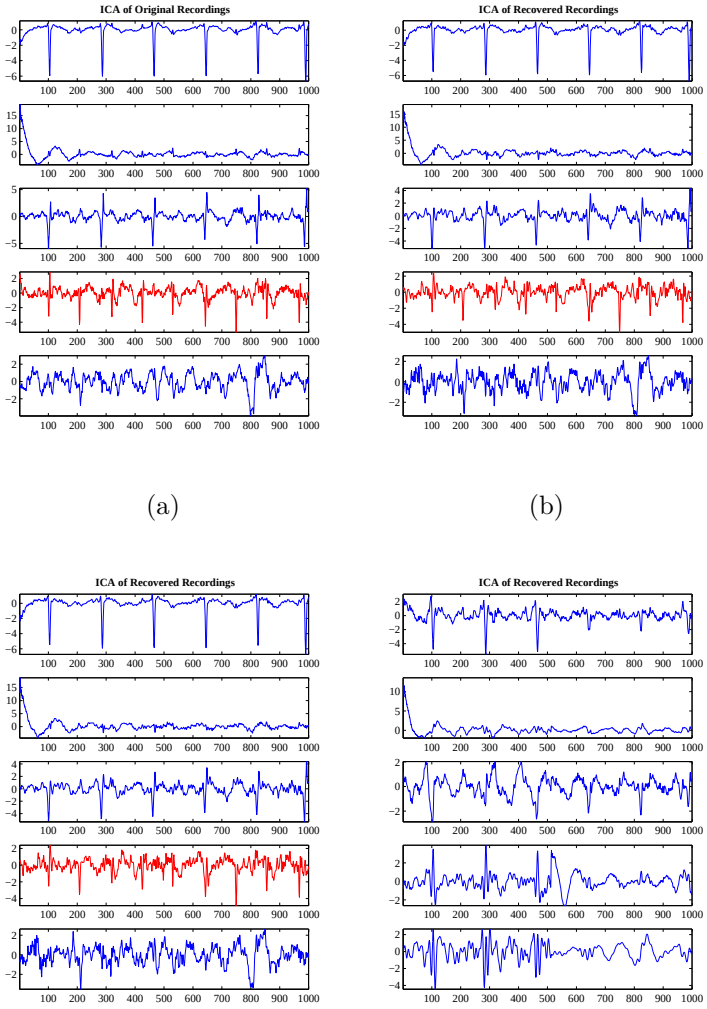

Figure 4: ICA decomposition of the original dataset (a), the recovered dataset by BSBL-FM(2) (b), BSBL-FM(1) (c) and VBGS (d) (only the first 1000 sampling points of the datasets are shown). The fourth ICs in (a), (b) and (c) are the extracted FECGs from the original dataset and the reconstructed datasets. The ICA decomposition on the recordings recovered by VBGS (d) didn't extract the FECG.

References

[1] E. Candes, M. Wakin, An introduction to compressive sampling, Signal Processing Magazine, IEEE 25 (2008) 21 –30.

[2] Z. Zhang, B. D. Rao, Extension of SBL algorithms for the recovery of block sparse signals with intra-block correlation, to appear in IEEE Transactions on Signal Processing (2012).

[3] M. Yuan, Y. Lin, Model selection and estimation in regression with grouped variables, J. R. Statist. Soc. B 68 (2006) 49–67.

[4] R. G. Baraniuk, V. Cevher, M. F. Duarte, C. Hegde, Model-based compressive sensing, IEEE Transactions on Signal Processing 56 (4) (2010) 1982–2001.

[5] S. D. Babacan, S. Nakajima, M. N. Do, Bayesian group-sparse modeling and variational inference, Submitted to IEEE Transactions on Signal Processing (2012).

[6] Z. Zhang, B. D. Rao, Sparse signal recovery with temporally correlated source vectors using sparse bayesian learning, IEEE Journal of Selected Topics in Signal Processing 5 (2011) 912–926.

[7] Z. Zhang, T.-P. Jung, S. Makeig, B. D. Rao, Compressed sensing for energy-efficient wireless telemonitoring of non-invasive fetal ECG via block sparse Bayesian learning, IEEE Transaction on Biomedical Engineering 60 (2) (2013) 300–309.

[8] M. E. Tipping, A. C. Faul, Fast marginal likelihood maximisation for sparse bayesian models, in: C. M. Bishop, B. J. Frey (Eds.), Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics, Key West, FL, pp. 3–6.

[9] Y. C. Eldar, P. Kuppinger, H. Bolcskei, Block-sparse signals: uncertainty relations and efficient recovery, IEEE Transaction on Signal Processing 58(6) (2010) 3042–3054.

[10] S. Ji, Y. Xue, L. Carin, Bayesian compressive sensing, IEEE Transactions on Signal Processing 56 (6) (2008) 2346–2356.

[11] E. Van Den Berg, M. Friedlander, Probing the pareto frontier for basis pursuit solutions, SIAM Journal on Scientific Computing 31 (2008) 890–912.

[12] D. Donoho, J. Tanner, Observed universality of phase transitions in high-dimensional geometry, with implications for modern data analysis and signal processing, Philosophical Transactions of the Royal Society A 367 (2009) 4273–4293.

[13] H. Mamaghanian, N. Khaled, D. Atienza, P. Vandergheynst, Compressed sensing for real-time energy-efficient ecg compression on wireless body sensor nodes, Biomedical Engineering, IEEE Transactions on 58 (2011) 2456–2466.

[14] A. Hyvarinen, Fast and robust fixed-point algorithms for independent component analysis, Neural Networks, IEEE Transactions on 10 (1999) 626–634.